From Reactive to Predictive Regulation in Metros

Karim Kecir, Loic Helouet, Pierre Dersin, Bruno Adeline, Andrea d'Ariano

From Reactive to Predictive Regulation in Metros

Karim Kecir[1,2] Loïc Hélouët[1] Pierre Dersin[2]
Bruno Adeline[2] Andrea D'Ariano[3]
[1]ALSTOM, [2]INRIA Rennes, FRANCE [3]UNIV. ROMA 3, ITALY
{karim.kecir,loic.helouet}@inria.fr {pierre.dersin,bruno.adeline}@alstom.com
a.dariano@ing.uniroma3.it

Traffic regulation is an important part of metro networks. Trains in an urban network need guidance to ensure smooth operation of the system, passengers satisfaction, and also to meet criteria fixed by operators or by their clients (usually quality contracts are fixed by local authorities). These criteria can vary from a line to another, but also depending on the day and time of the day. Examples of criteria that have to be met are punctuality, regularity of service, energy consumption...

For these systems, forecasts are designed in the form of timetables describing departure and arrival dates of trains at stations, or with promises of regular pace at some stations. However, due to unpredictable delays, these plannings are never met. Such random delays can originate from weather conditions, users misbehavior, signaling systems failure, etc. As a consequence, a metro network cannot be operated without corrective mechanisms called *regulation algorithms*.

Currently, regulation algorithms are mainly *event-based* and *reactive*: upon arrival of a train, the difference between the forecast reference arrival date and the actual observed arrival date calls for corrections to the forecast: next departures can be delayed, dwell times can be shortened, commercial speeds can be increased and, in some cases, trains ordering can even be modified. These algorithms are mainly application of rules of the form `"if train` x `is late by more than` y `seconds, then shorten dwell time by` v `seconds"`. Of course, algorithms can be more involved than in this example, and may consider more parameters than a single train delay; yet, they remain quite local decisions. In this setting, one can notice that regulation advices are just application of rather logical rules aiming at recovering delays, but whose optimality is not certain. Experience shows nevertheless that these systems work well in practice and are sufficient to recover from small delays. Nowadays, as the number of daily commuters increases, metro systems have to face growing traffic, while maintaining a high quality of service. In addition to service increase, energy consumption optimization is also becoming a central concern in most cities. There is hence a clear need for algorithms with optimal performance, and for tools to demonstrate this optimality.

We propose a framework to model networks that integrate optimization schemes in their regulation, and tools to evaluate the performance of these optimized regulation algorithms. The main idea here is to consider more global decisions: instead of reacting to a local delay, we advocate the fact that solutions returned by regulation algorithms have to consider elements from the whole network and its planned forecast. We also advocate the fact that decisions that are proposed by regulation algorithms have to be optimized: instead of computing a rescheduling that simply postpones forecasts depending on a measured delay and on fixed thresholds, regulation can:

- reconsider ordering of trains (and even insert or remove trains from a network)
- search solutions that optimize some criteria (mean delay, energy consumption,...) a priori, at least for a bounded time window

This setting is *predictive*, as one reconsiders for a bounded time windows the whole forecast ordering and timing. It tries to be optimal, by returning the best solution for given time window. Note here that one cannot expect a regulation algorithm to return the best possible solution as, first, optimization algorithms use heuristics to solve problems in decent time; and, second, even if an optimal solution is proposed for a given setting, disturbances can still occur thus invalidating a planning that has been recomputed, and which will hence not be optimal in this modified setting.

We are developing the SIMSTORS tool for the analysis of regulation algorithms performances. This tool is designed as follows. It is decomposed into three parts: the first component is a simulator that represents a network, animates trains and introduces randomness symbolizing disturbances. The underlying model for networks is a variant of stochastic Petri nets, and is detailed in [3]. The second part is a schedule representing the predicted timetable, and the last part is a regulation algorithm. The simulator part models trains moves, accepts inputs from the regulation part, and plans departures as expected from the schedule. In turns, it provides information to the regulation algorithm on arrival and departure dates of trains. The schedule is part of the input data of the system and evolves during the simulation. The regulation part is an actual algorithm running in a real system. It receives occurrence dates for departures and arrivals, and returns rescheduling decisions (mainly delaying next events). The overall output of this system gives a log of the simulated period, that can be analyzed. Stochastic simulation campaigns (à la Monte-Carlo) allow to derive statistics, and measure efficiency of regulation algorithms.

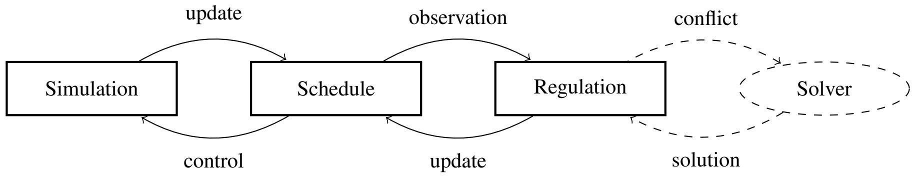

Figure 1: Coupling of the regulation evaluation simulator with a conflict solver

We propose to extend this simulation scheme to evaluate integration of optimization techniques in regulation. The main idea is, upon occurrence of an event (mainly departures and arrivals), apply regulation techniques whose rescheduling uses the output of an optimization algorithm. There are constraints for this new regulation scheme. First of all, the optimization algorithm has to return a fast enough answer; this means that optimization has to be performed for a bounded time window. This is not a real disadvantage as optimal solutions for large time windows have high chances to be reconsidered. This solution also has to provide fast reactive decisions for events that occur during computation time of the optimizer (optimization applies only to the next events). This means that our regulation algorithm has to be an hybrid solution applying fast reactive solutions for early events, and quasi optimal solutions to remaining events occurring after the delay needed to compute optimized solutions. The last constraint in this setting is that the optimizer has to be fast enough to allow for Monte-Carlo simulation. The optimizer that we plan to use is AGLIBRARY [1, 2]. It is a solver designed based on a set of OR models and algorithms for the resolution of conflicts (situations where several trains require the same portion of a network) in complex rail systems.

References

[1] Samà M., D'Ariano A., Corman F., Pacciarelli D. A variable neibourhood search for fast train scheduling and routing during disturbed railway traffic situations. Computers & Operations Research, 78, 480-499, 2017.

[2] D'Ariano A., Samà M., D'Ariano P., Pacciarelli D. Evaluating the applicability of advanced techniques for practical real-time train scheduling. Transportation Research Procedia, 3, 279-288, 2014.

[3] Hélouët L., Kecir K.: Realizability of Schedules by Stochastic Time Petri Nets with Blocking Semantics. Petri Nets 2016, 155-175, 2016.